1. System deploy

You should ensure the following things are in your computer

System: Linux (recommended)

Software: Spark (with python API pyspark)

Python Packages: h5py, matplotlib

2. Install

2.1 Decompress the source code folder, ensure all the code are in one folder so that they can import each other.

2.2 Open the file “pathList.py”, find the “path” class. Change the value of “pathToDataset” to the path of your whole dataset. Change the value of “pathToPathList” to the path where you want to create the pathList.txt.

2.3 Run pathList.py to generate the pathList.txt file.

3. Files

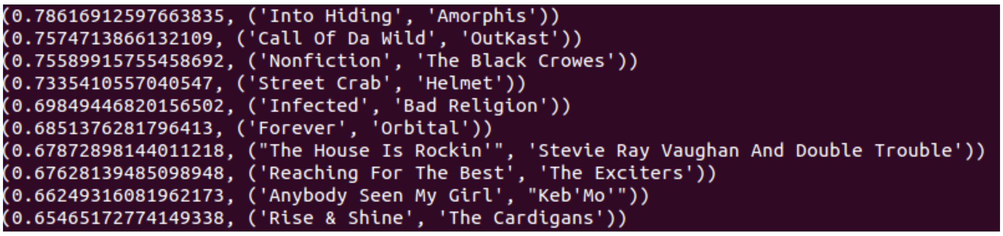
Each file implements one simply function. They are not depend on each other but all depend on the pathList.txt. Therefore please ensure that you have generated the pathList.txt in last step before running each file.

3.1 yearForTop10Songs.py

Use: show the most 10 popular songs’ name and their artists’ name for a specific year. A float number in 0 – 1 represents the hotness of the song. Larger the number is more popular the song is.

Run: you need to pass one argument as the year you want to check. E.g. `pyspark yearForTop10Songs.py 1994`

Result should be like this:



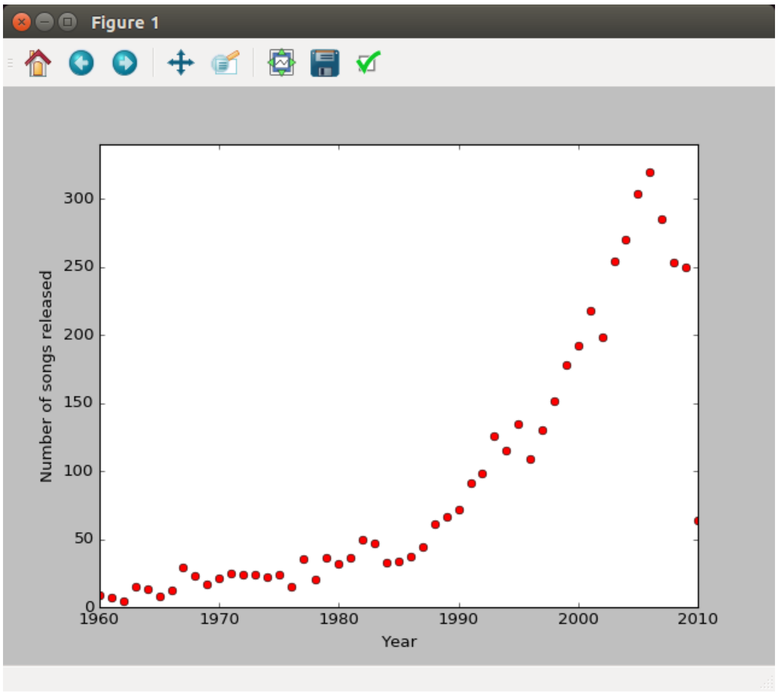
```
(0.78616912597663835, ('Into Hiding', 'Amorphis'))
(0.7574713866132109, ('Call Of Da Wild', 'OutKast'))
(0.75589915755458692, ('Nonfiction', 'The Black Crowes'))
(0.7335410557040547, ('Street Crab', 'Helmet'))
(0.69849446820156502, ('Infected', 'Bad Religion'))
(0.6851376281796413, ('Forever', 'Orbital'))
(0.67872898144011218, ("The House Is Rockin'", 'Stevie Ray Vaughan And Double Trouble'))
(0.67628139485098948, ('Reaching For The Best', 'The Exciters'))
(0.66249316081962173, ('Anybody Seen My Girl', "Keb'Mo'"))
(0.65465172774149338, ('Rise & Shine', 'The Cardigans'))
```

3.2 yearAmount.py

Use: show the number of music released in each year without missing value mixed

Run: Simply run it by typing `pyspark yearAmount.py`

Result should be like this:

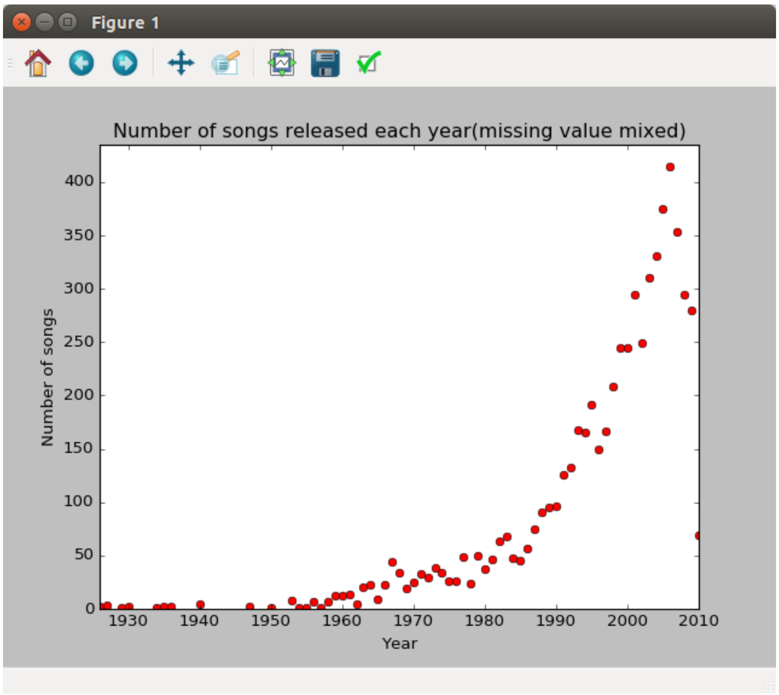


3.3 yearForMusicReleased_FixMissingValue.py

Use: show the number of music released in each year with missing value mixed

Run: Simply run it by typing `pyspark yearForMusicReleased_FixMissingValue.py`

Result should be like this:



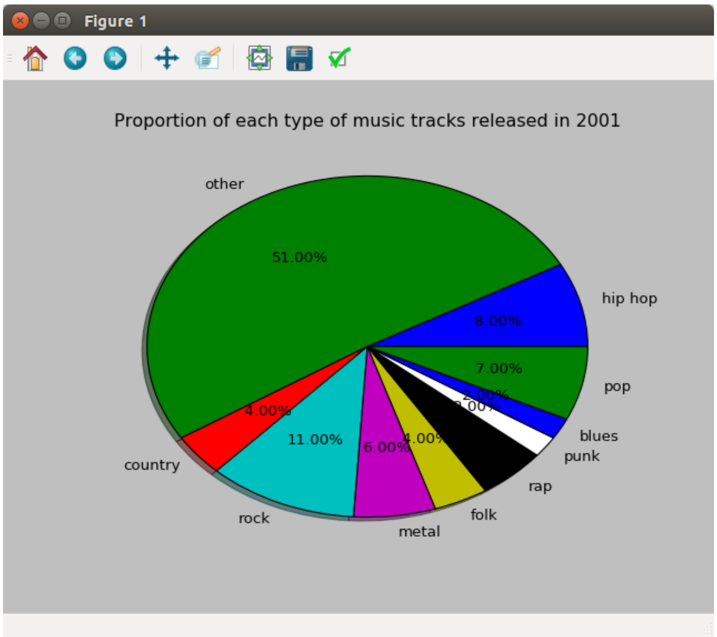
3.4 yearForMusicTagProportion.py

Use: show the proportion of different type of music in a specific year

Run: you need to pass one argument as the year you want to check. E.g. `pyspark`

```
yearForMusicTagProportion.py 2001
```

Result should be like this:



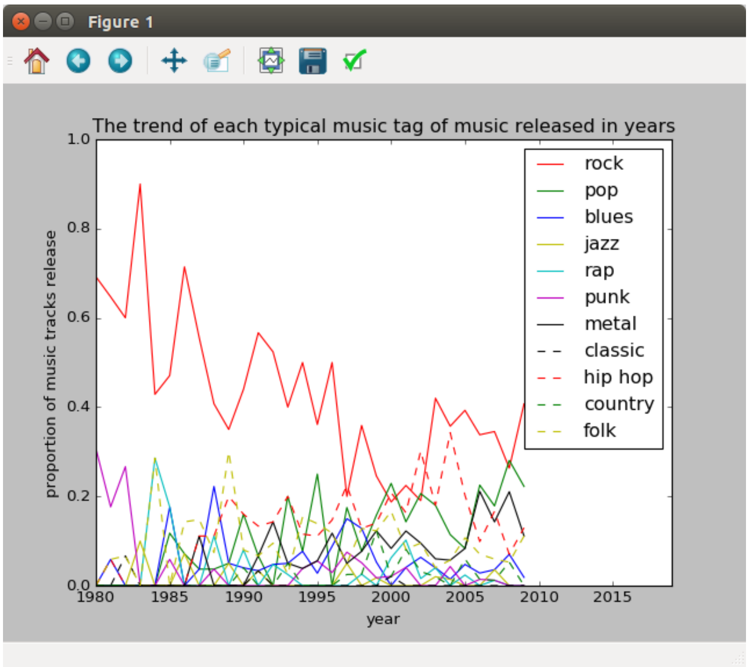
3.5 yearForMusicTagTrend.py

Use: show the trend of proportion of different types of music in a specific year range

Run: you need to pass tow arguments as the range of years you want to check. E.g.

```
pyspark yearForMusicTagTrend.py 1980 2009
```

Result should be like this:

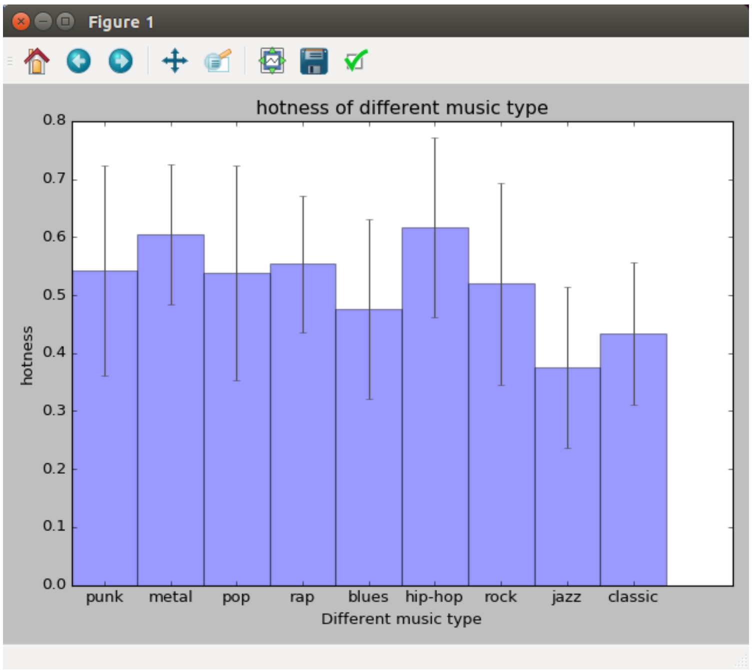


3.6 musicInfo.py

Use: show tempo, loudness or hotness for each music type. A bar stands for the mean value of that type of music, and the short line on the bar is its standard deviation.

Run: you need to pass one argument as the type of information you want to check. It can only be “tempo”, “loudness” or “hotness”. E.g. `pyspark  musicInfo.py hotness`

Result should be like this:



3.7 musicTrend.py

Use: show the trend of tempo, loudness or hotness for a specific music type or all the music types. Each red point is a piece of music data.

Run: you need to pass two arguments as input. The first one is the type of music you want to check, such as “rock”, “rap”. If you want to check all kinds of music, type “general”. The second one is the type of information. It can only be “tempo”, “loudness” and “hotness”. E.g. `pyspark musicTrend.py rock hotness`

Result should be like this:

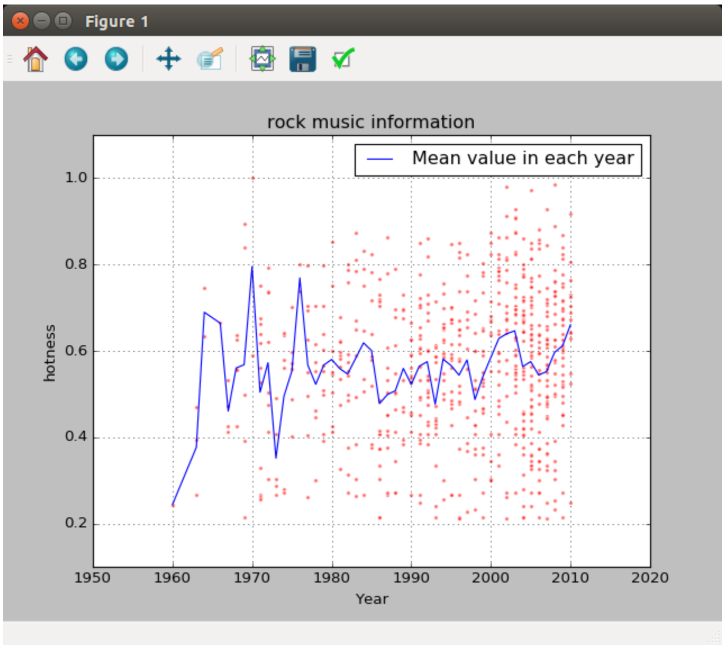


Figure 1
rock music information
Mean value in each year
1.0
0.8
0.6
0.4
0.2
hotness
1950
1960
1970
1980
1990
2000
2010
2020
Year